Meridian Administration Tools

Alarm Management

User Guide

Document Number: P0906946
Document Release: Standard
Date: April 2000

Printed in the United States of America

Information is subject to change without notice. Nortel Networks reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant. This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC rules, and the radio interference regulations of Industry Canada. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at their own expense.

Revision history

Date	Document Version	Product Release
April 2000	Standard	6.6
June 1999	Standard	6.5
March 1999	Standard	6.4
December 1998	Standard	6.1
June 1998	Standard	6
July 1997	Standard	5.7

Contents

List of figures

List of tables

Alarm Management

Alarm Management consists of a number of components to improve handling of system-generated alarms and events. Alarm Management is only available for Meridian 1 systems running X11 Release 22 or later and configured with the X11 MAT package (296).

Alarm Banner

The Alarm Banner alerts you to the number of new critical, major, and minor alarms in the system event log. A new critical alarm will cause the critical alarm count in the Alarm Banner to flash. The Alarm Banner window also provides the count of new alarms.

If you wish to learn more about an alarm, click the Events button to launch the Events window. If there are no alarms, you can log out or leave the Alarm Banner displayed and go on to another task. See “Alarm Banner” on page 17.

Events window

The Events window displays the Meridian 1 system's Event Log, allowing you to view all recent system alarms and events (previously stored in the Meridian 1 history file). The Events window displays active events in a way that lets you quickly view the most important events. System events with a severity of critical, major, or minor are considered alarms—alarms are events which may require some corrective action. System events with a severity of Info are for informational purposes only and are not considered alarms.

> ***Note:*** The Event Log is preserved through a sysload and initialization of a Meridian 1.

The Events window allows you to view, acknowledge, and mark alarms as cleared, and view alarm definitions. You can easily define filters with checkboxes to determine the severity of events that appear in the list. Once you have fixed a problem, you can mark that alarm as cleared in the list. See “Events Window” on page 21.

Event Default Table and Event Preference Table

The Event Default Table specifies the default severity for events. See “Changing alarm severity or escalation” on page 28. The Event Preferences window allows you to modify these severity defaults and to define escalation and suppress thresholds for each event. See “Creating and customizing Event Preferences” on page 28.

Alarm Notification

The Alarm Notification application expands upon the existing Alarm Management application by providing capabilities to send various notifications when specific events occur. Using advanced rule-based event filtering capabilities, the Alarm Notification application complements existing external alarm management systems by adding greater flexibility, features, and capacity. The Alarm Notification application receives Simple Network Management Protocol (SNMP) events from specified systems and sends out alarm notifications when specified events are detected.

Alarm processing and notification is defined in a script file which the user sets up. You can create a new script file or modify the sample script file included with the application. Be sure to work with a copy of this sample file to preserve the original. Users should be familiar with scripting logic and programming principles to effectively use and extend this application’s capabilities.

Conventions used in this guide

This guide uses the following typographical conventions:

- User input—**This font** alerts you to information that you enter on your keyboard; or, using your mouse, this font indicates buttons to click or menu selections to make.
- Multi-lettered keys—Angle brackets denote a single multi-lettered key on your keyboard. For example, **<Esc>** denotes the Escape key, labeled *Esc* on PC keyboards.
- Key sequences—Keys that you press at the same time include at least one multi-lettered key and are not separated by spaces in text. For example, **<Alt>V** instructs you to press the **<Alt>** and **V** keys at the same time.
- Hot keys—You can access menu commands by using the mouse or your keyboard. Menu items show one letter as underlined. To choose a menu item from the keyboard, hold down the **<Alt>** key and press the underlined key. For example, press **<Alt>F** to open the **File** menu.
- *Windows* refers to the Microsoft family of graphical user interface (GUI)-based operating systems.

Alarm Banner

Before MAT 5 was introduced, you had to determine system status by reviewing the history file to look for problems, and issuing a number of status commands in various overlays to look for disabled or faulty hardware. The Alarm Banner window now automatically alerts you to this information in a simple, direct manner.

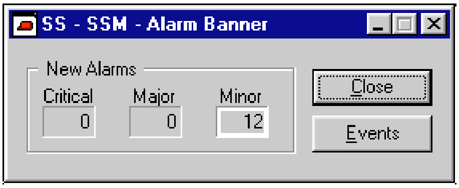
The Alarm Banner window alerts you to new critical alarms and gives you the count of all new alarms. If you wish to learn more about an alarm, press the **Events** button to launch the Events window. If there are no alarms, you can log out or leave the Alarm Banner displayed and go on to another task.

When a new critical alarm arrives, the system beeps if the notification option has been set, and the Alarm Banner title bar icon and Events window task bar icon flash. The flashing continues until you click anywhere in the Alarm Banner or Events window.

Launching the Alarm Banner

From the **Maintenance** menu of the System window, select **Alarm Banner**. Alternately, you can double-click the **Alarm Banner** icon in the System window toolbar. The Alarm Banner window appears as shown in Figure 1.

Figure 1
Alarm Banner window



SS - SSM - Alarm Banner

New Alarms

Critical	Major	Minor
0	0	12

The Alarm Banner keeps you informed about the current status of the system. You might typically check the current status of the system for the following reasons:

- standard operating procedure (for example, check every morning)
- investigating a suspected or reported problem
- checking and monitoring the system while performing other tasks

Launching the System Events window

You can launch the System Events window from the Alarm Banner by clicking the **Events** button.

Alarm Banner notifications

The primary function of the Alarm Banner is to notify you when a new alarm arrives in the following ways:

- The outline of the field holding the alarm count flashes to indicate the arrival of a new critical alarm.
- The event count in the Alarm Banner is incremented appropriately for all users.
- One or more beeps sound. This is optional. See “Creating an Event Preference definition” on page 31.

  Note: The Events window must be active for the beep to sound.
- If the Alarm Banner has been closed by the user, the sound notification is provided by the Events window. The window is not brought to the front, as this may interfere with your current task.

There is no alarm notification beep if the sound has been disabled. The count field outline still flashes and the count increments to indicate the arrival of a new alarm.

Events Window

The Events window displays recent system alarms and events. The Events window displays active events in a way that lets you quickly view the most important events.

The Events window allows you to view, acknowledge, and mark alarms as cleared, and view event preferences. (System events with a severity of critical, major, or minor are considered alarms.) Once you have fixed a problem, you can mark that alarm as cleared in the list.

Using the Events window

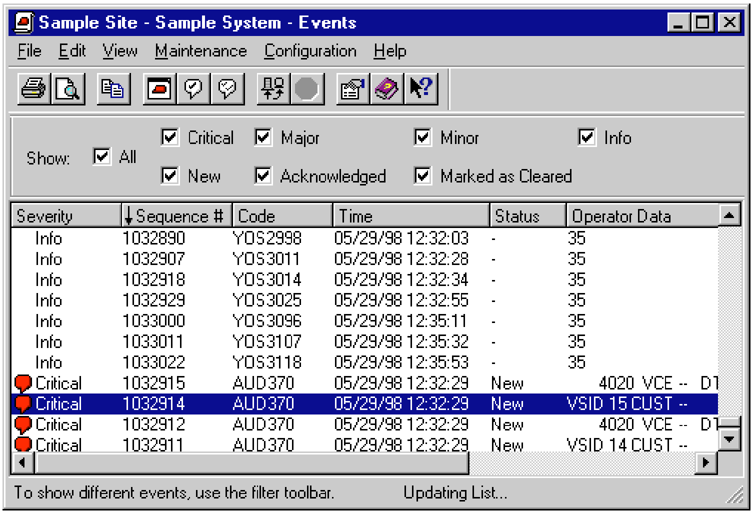
Once you open the Events window, you can perform the following tasks:

- get a description of an event
- acknowledge an alarm you intend to clear—this communicates your intention to others who may be working on the system
- locate an alarm in the Event file to identify the cause of the problem
- learn more detail about an alarm
- mark as cleared in the list after you have corrected the problem
- change system event preferences for all subsequent alarms
  - the severity of the alarm
  - the escalation threshold for an alarm type

Launching the Events window

From the **Maintenance** menu of the System window, select **Events**. If you are not connected to the Meridian 1, the connection login window appears to allow you to connect. Once you have connected, the Events window appears as shown in Figure 2.

Figure 2
Events window



Show: ☑ All ☑ Critical ☑ Major ☑ Minor ☑ Info

☑ New ☑ Acknowledged ☑ Marked as Cleared

Severity	Sequence #	Code	Time	Status	Operator Data
Info	1032890	YOS2998	05/29/98 12:32:03	-	35
Info	1032907	YOS3011	05/29/98 12:32:28	-	35
Info	1032918	YOS3014	05/29/98 12:32:34	-	35
Info	1032929	YOS3025	05/29/98 12:32:55	-	35
Info	1033000	YOS3096	05/29/98 12:35:11	-	35
Info	1033011	YOS3107	05/29/98 12:35:32	-	35
Info	1033022	YOS3118	05/29/98 12:35:53	-	35
Critical	1032915	AUD370	05/29/98 12:32:29	New	4020 VCE -- D1
Critical	1032914	AUD370	05/29/98 12:32:29	New	VSID 15 CUST --
Critical	1032912	AUD370	05/29/98 12:32:29	New	4020 VCE -- D1
Critical	1032911	AUD370	05/29/98 12:32:29	New	VSID 14 CUST --

To show different events, use the filter toolbar. Updating List...

The Events window provides a list of events and a menu bar from which you can learn more about the events. An event remains displayed in the list until expired by the system. The column headings identify and describe the events. To act on an event (such as acknowledging or marking an alarm as cleared) see “Getting around in the Events window” on page 23.

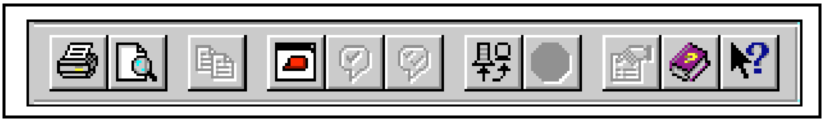
Getting around in the Events window

The following tools are available to you:

Using the Events toolbar

Use the Events toolbar to perform many of the commands in the Events window menus. Each button in the toolbar is documented in **What's This** help.

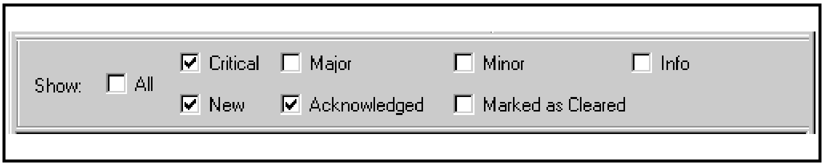
Figure 3
Events Toolbar



Using the Filter toolbar

Use the Filter toolbar to select the types of alarms or events that you wish to display in the list. Figure 4 shows the Filter toolbar. Click to check (or uncheck) an item. For example, you would check **New** and **Critical** to restrict the list to alarms that are new and critical. You can drag the Filter Toolbar to the top or bottom of the window.

Figure 4
Filter toolbar



Show: □ All ☑ Critical □ Major □ Minor □ Info
☑ New ☑ Acknowledged □ Marked as Cleared

Sorting the event list

Alarms and events are listed in order of occurrence (they are sorted according to the Time column). You can sort the list according to another column by clicking in that column heading. This is useful for prioritizing your work when dealing with a large number of alarms.

Click to sort in ascending order; click again for descending order. An arrow in the column heading indicates the current sort column and sort order.

Getting more detail on an alarm

If the data does not all fit in a column, you can resize a column by dragging the column divider. You can also use **Properties** in the **File** menu to display complete information about a selected event.

Double click on an alarm, or click the **What’s This** button to see reference information in Windows Help.

New critical alarms are indicated by the icon. The icon disappears when the alarm is acknowledged.

Using shortcuts

You can perform the following common tasks from the right mouse button popup menu:

- **Copy** an alarm: copies selected events to the clipboard (a temporary holding area). You can then insert the text into another document. Copy is unavailable when no text is selected.
- **Select All** alarms: selects all events in the list. You can print the selected text or copy it to the clipboard and insert it into another document.
- **Acknowledge** an alarm: changes the status of the selected alarm to Acknowledged. This informs other technicians that the problem is being investigated.
- **Mark as Cleared**: Changes the status of the selected alarm to Marked as Cleared. This informs other users that the problem is solved.
- Learn the **Properties** of an alarm: displays complete information about the selected event.
- **What’s This?** general help option: Changes the cursor to a “question mark” cursor and displays help on the next item you select.

Alarm descriptions

The Events window provides several columns of information about each active alarm. You can resize a column by dragging the column divider to make more room for text. Table 1 describes each column.

Table 1
Columns in the Events window

Column	Description
Severity	The alarm severity (critical, major, or minor) or a non-alarm event (information). An icon indicates an unacknowledged critical alarm.
Sequence	All events are given a unique number in the order they occur.
Code	A code that identifies the event. It includes the error category (for example NWS300) and a five-digit error number.
Time	The date and time that the alarm occurred.
Status	Current alarm status (appears with a dash “-” for non-alarms). New indicates an alarm has not been acknowledged or cleared. Acknowledged indicates an alarm in the process of being cleared. Marked as Cleared indicates the alarm has been manually cleared.
Operator Data	Data produced by the equipment that generated the event. Contents may vary. Typically includes a description of the event and the equipment affected (component ID information, such as the loop number or TN).

Events window menus

The Events window menu selections are designed to be fully documented in the on-line help. You can use **What’s This** help for any menu item or toolbar button.

Determining the cause of an alarm

An alarm may be caused by another system event, such as a BUG message. By examining the events immediately preceding an alarm, you may be able to isolate the source of the problem. Use the scroll bar to browse through the event list. To display help on a selected alarm, double-click the alarm.

Acknowledging an alarm that you intend to clear

You can acknowledge a new alarm to inform others that you will investigate the problem and clear the alarm. Your acknowledgment appears in the **Status** field of the Events window. The Events are updated for all MAT users.

> ***Note:*** Events with a status of **Info** cannot be acknowledged or marked as cleared. Alarms that have been marked as cleared cannot be acknowledged.

To acknowledge an alarm:

1 Select the desired alarm(s).

You can use the checkboxes at the top of the window to restrict the list to certain types of alarms. Use **<Shift>-click** to select a range of alarms. Use **<Ctrl>-click** to select multiple alarms. To select all alarms, choose **Select All** from the **Edit** menu.

2 Choose **Acknowledge** from the **Maintenance** menu or the right mouse button.

Once you acknowledge an alarm, the Status field for all selected alarms in the Events window is marked “Acknowledged.”

Marking an alarm as cleared

After you fix a problem, you will typically mark the associated alarm as cleared. The term *Mark as Cleared* is used because clearing an alarm only changes its status—it does not actually fix the problem.

To mark alarms as cleared:

1. Select one or more alarms in the Events window. You can use the checkboxes at the bottom of the window to restrict the list to certain types of alarms. Use **<Shift>-click** to select a range of alarms. Use **<Ctrl>-click** to select multiple alarms. To select all alarms, choose **Select All** from the **Edit** menu.

   Note: You can usually save time by displaying the type of alarm of interest using the Filter bar before selecting individual alarms.

2. Select **Mark as Cleared** in the **Maintenance** menu, and click **OK** to confirm.

Acknowledging and clearing alarms is optional. You can clear alarms without first acknowledging them. If you do not clear alarms, the oldest alarms are expired by the system when it reaches the maximum number of alarms.

> ***Note:*** It is recommended that you clear alarms as you fix problems so that the Events window accurately reflects the state of the system. Events with a status of **Info** cannot be acknowledged or marked as cleared. Alarms that have been marked as cleared cannot be acknowledged.

When you clear an alarm, the following happens:

- The Alarm Status field for all selected alarms is updated in the System Event File with “Marked as Cleared”
- The counts in the Alarm Banner are adjusted appropriately for all users.

Learning more detail about an alarm

You can use the following methods to learn more about selected alarms:

- Select **Properties** in the **File** menu or click the **Properties** button to see all information for the selected alarm.
- Double click on an alarm to see reference information in Windows Help.

Changing alarm severity or escalation

Use the **Event Preferences** command in the **Configuration** menu to specify the severity of events (critical, major, minor, or event). These options are set on a per-system basis. The system uses an Event Default Table which predefines the severity of all events. Typically, you modify these settings only when installing or upgrading the system. See “Creating and customizing Event Preferences” on page 28.

Viewing the Event Default Table

The Event Default Table contains the default severity settings of all system events. Use the table to verify default settings before adding event preferences.

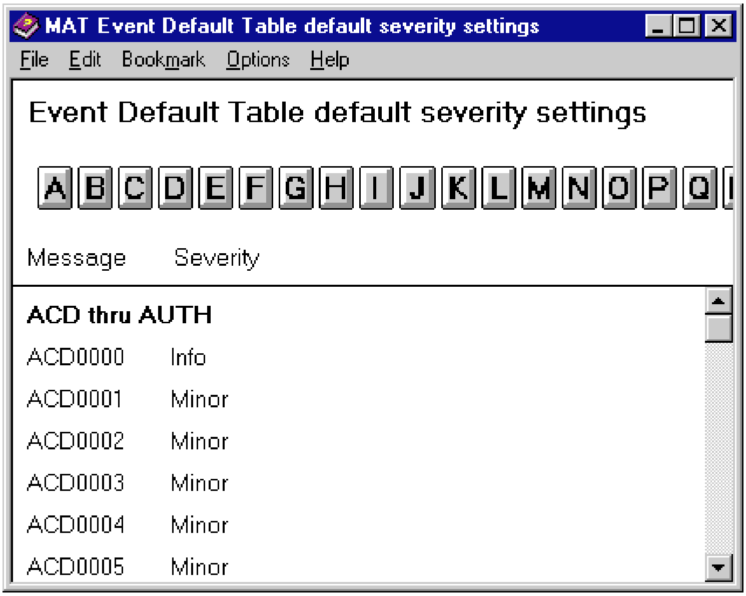
To display the Event Default table, choose **Event Preferences** from the **Configuration** menu, then choose **Event Default Table** from the **Help** menu in the Event Preferences window. See Figure 5.

Creating and customizing Event Preferences

You can customize alarms for this system by changing the default alarm severity and escalation threshold using the Event Preferences window. See Figure 6. The escalation setting defines the maximum number of times an event can occur within a defined period of time (the global time window) before it escalates to the next higher level of severity. For example, if you set escalation to “10 occurrences in 1 minute” for a minor alarm, the alarm will escalate to a major alarm when it occurs more that 10 times within a 1 minute period. See “Creating an Event Preference definition” on page 31 for steps describing how to set escalation parameters.

To open the Event Preferences window, select **Event Preferences** from the **Configuration** menu in the Events window.

Figure 5
A portion of the Event Default Table window



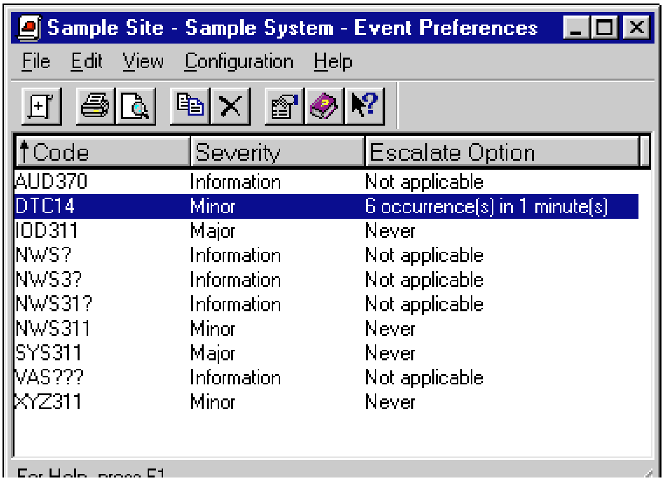
MAT Event Default Table default severity settings
File Edit Bookmark Options Help
A B C D E F G H I J K L M N O P Q

Event Default Table default severity settings

Message	Severity
ACD thru AUTH	
ACD0000	Info
ACD0001	Minor
ACD0002	Minor
ACD0003	Minor
ACD0004	Minor
ACD0005	Minor

Before changing an alarm definition, you may wish to look up the default settings in the Event Default Table. See “Viewing the Event Default Table” on page 28.

Figure 6
Event Preferences window


Sample Site - Sample System - Event Preferences
File Edit View Configuration Help
For Help, press F1

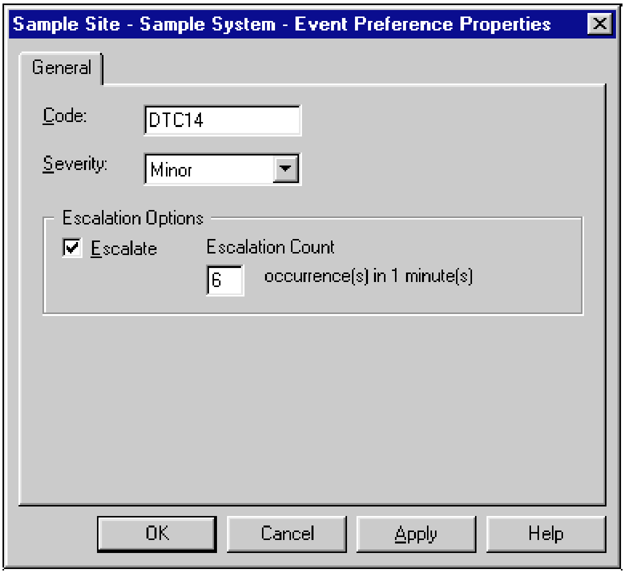
Code	Severity	Escalate Option
AUD370	Information	Not applicable
DTC14	Minor	6 occurrence(s) in 1 minute(s)
IOD311	Major	Never
NWS?	Information	Not applicable
NWS3?	Information	Not applicable
NWS31?	Information	Not applicable
NWS311	Minor	Never
SYS311	Major	Never
VAS???	Information	Not applicable
XYZ311	Minor	Never

Creating an Event Preference definition

1 Choose **Event Preferences** from the **Configuration** menu—the Event Preferences window appears.

2 Choose **Add Event Preference** from the **Configuration** menu. The **Event Preference Properties** sheet appears—see Figure 7.

Figure 7
Event Preference Properties sheet



Sample Site - Sample System - Event Preference Properties

General

Code: DTC14

Severity: Minor

Escalation Options

☑ Escalate

Escalation Count

6 occurrence(s) in 1 minute(s)

3 In the **Code** field, type the alarm or event ID.

The ID includes the event category (such as **BUG**, or **NWS**) and the five-digit event number.

You may use the wildcard symbol **?** to represent a group of error code numbers. For example, **NWS3??** represents all error codes between NWS300 and NWS399.

4 To change the alarm severity, press the down arrow in the **Severity** field and choose a setting from the popup menu.

5 To change the escalation threshold, check the **Escalate** box. Type a number in the **Escalation** field.

The escalation setting defines the maximum number of times an event can occur within a defined period of time (the global time window) before it escalates to the next higher level of severity. For example, of you set escalation to “10 occurrences in 1 minute” for a minor alarm, the alarm will escalate to a major alarm when it occurs more that 10 times within a 1 minute period.

The global time window is set in the Meridian 1, and can range from 0 to 60 seconds.

6 Click **OK**.

Alarm Notification

The Alarm Notification application uses the existing MAT architecture to connect to Meridian 1 systems and other supported systems and equipment which can generate SNMP events to detect specified events. For Meridian 1 systems, the X11 SNMP Open Alarms Package (315) must be present and activated along with the packages required for MAT.

How it works

The Alarm Notification application receives SNMP events from designated network equipment over an Ethernet network and sends out alarm notifications when specified event conditions are detected. Received events are examined against a set of programmed rules which may activate notifications of different types. These notifications include:

- SNMP traps or events transmitted to predefined destinations
- text notification over a modem
- pager notification to alpha or numeric pagers
- electronic mail using Simple Mail Transfer Protocol (SMTP)
- log file

SNMP events are displayed at the MAT PC in the Alarm Notification window. You can also view events with a web browser connected to a configured web server. When the application starts, three application control files are loaded: a devices file, a configuration file, and a scripts file.

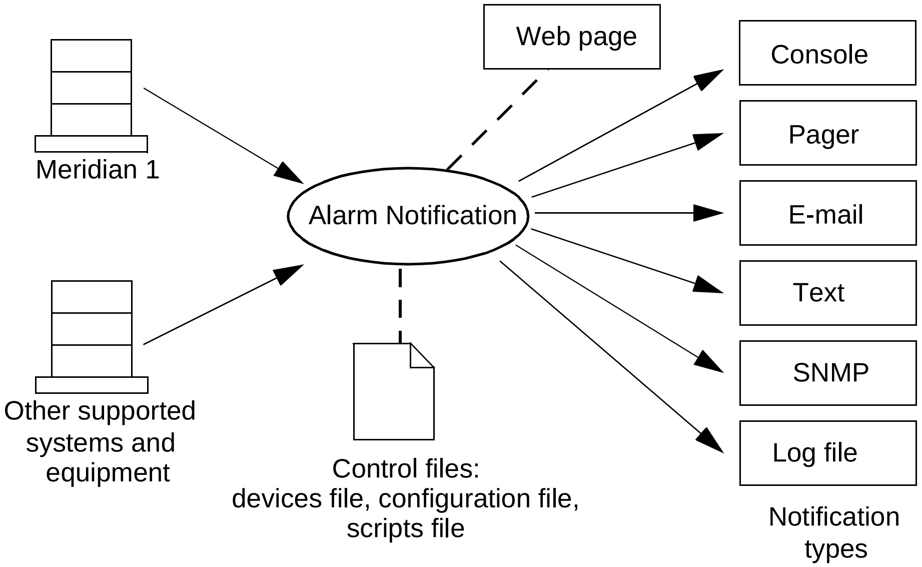
> ***Note:*** These control files must be present and configured for the Alarm Notification application to work correctly. See “Setting up Alarm Notification” on page 35.

The devices file specifies the SNMP devices to be monitored. The devices file must be defined before you can start alarm monitoring. A sample devices file, Devices.txt, is provided in the MAT directory.

The configuration file defines event information (SNMP traps) to be monitored from the SNMP devices specified in the devices file. Event values are mapped to variable names which can be used in the scripts file. The configuration file must be defined before you can start alarm monitoring. A sample configuration file, Config.txt, is provided in the MAT directory.

The scripts file defines how alarms are processed and which notifications are used.

The diagram below shows a functional overview of the application.


Web page
Console
Pager
E-mail
Text
SNMP
Log file
Notification types
Meridian 1
Alarm Notification
Other supported systems and equipment
Control files: devices file, configuration file, scripts file

A scripting language is included within the application to allow users to define alarm processing and notifications. An external text editor is required for creating scripts.

Use the scripting language to:

- define new SNMP device types and devices to be recognized by the Alarm Notification application
- define how to process additional events
- define new responses and notification types to predefined events

Note: Events from undefined systems or devices are ignored.

Setting up Alarm Notification

Before Alarm Notification can function correctly, control files must be set up first. Control files include the devices file, the configuration file, and the scripts file.

The following lists an overview of the procedures to follow to set up Alarm Notification.

1 Make sure you have the control files correctly installed and Run Options defined.

2 Determine the IP address of your MAT PC on which you will view the events. This PC must have Alarm Notification correctly installed. The PC must be networked with the system to be monitored.

3 Use Overlay 117 to enable alarms to be sent to your MAT PC.

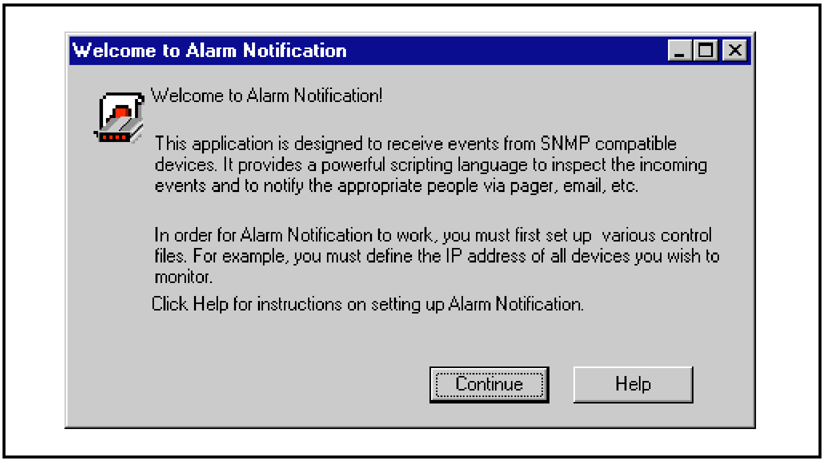
Install Alarm Notification control files

Make sure you have the control files correctly installed. Control files define which systems are monitored and which events are processed. For detailed instructions for defining Run Options, consult the online Help. Once the control files are defined, click on the box marked “Auto-start scripts on program launch” under the General tab in Run Options to automatically load these scripts when you next launch the application. See Figure 14

The first time the application is launched, the following welcome dialog box appears. Subsequent sessions do not display this window. See Figure 8.

Click **Help** to view the detailed instructions for setting up the control files. See Figure 9. Click **Continue** to go directly to the Alarm Notification window. See Figure 11.

Figure 8
Welcome to Alarm Notification window



Welcome to Alarm Notification!

This application is designed to receive events from SNMP compatible devices. It provides a powerful scripting language to inspect the incoming events and to notify the appropriate people via pager, email, etc.

In order for Alarm Notification to work, you must first set up various control files. For example, you must define the IP address of all devices you wish to monitor.

Click Help for instructions on setting up Alarm Notification.

Devices file

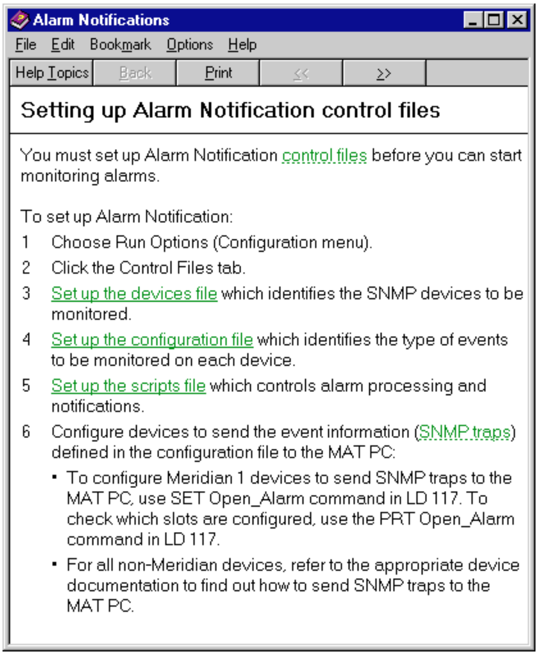
The *devices file* contains the list of monitored systems. Monitored systems must be previously defined in the other configuration file using the word *device*. Users can add reference information to monitored systems specifying:

- the IP address of the system or its name from the PC hosts file
- an alias for any system name or IP address

> ***Note:*** Within the Alarm Notification application, systems can be referenced by the specified alias.

Before you can start alarm monitoring, you must set up a devices file by renaming a copy of the sample Devices.txt file, then add IP addresses for the devices you want to monitor. See “Control Files Included with Alarm Notification” on page 71.

Figure 9
Help for setting up control files



Setting up Alarm Notification control files

You must set up Alarm Notification control files before you can start monitoring alarms.

To set up Alarm Notification:

1 Choose Run Options (Configuration menu).

2 Click the Control Files tab.

3 Set up the devices file which identifies the SNMP devices to be monitored.

4 Set up the configuration file which identifies the type of events to be monitored on each device.

5 Set up the scripts file which controls alarm processing and notifications.

6 Configure devices to send the event information (SNMP traps) defined in the configuration file to the MAT PC:

- To configure Meridian 1 devices to send SNMP traps to the MAT PC, use SET Open_Alarm command in LD 117. To check which slots are configured, use the PRT Open_Alarm command in LD 117.
- For all non-Meridian devices, refer to the appropriate device documentation to find out how to send SNMP traps to the MAT PC.

To set up the devices file:

1 In the Windows Explorer, rename a copy of the sample Devices.txt file, located in the MAT directory:

```
C:\Nortel\Common Data\Alarm Notification\Control Files
```

For example, the new filename might be *my_devices*.

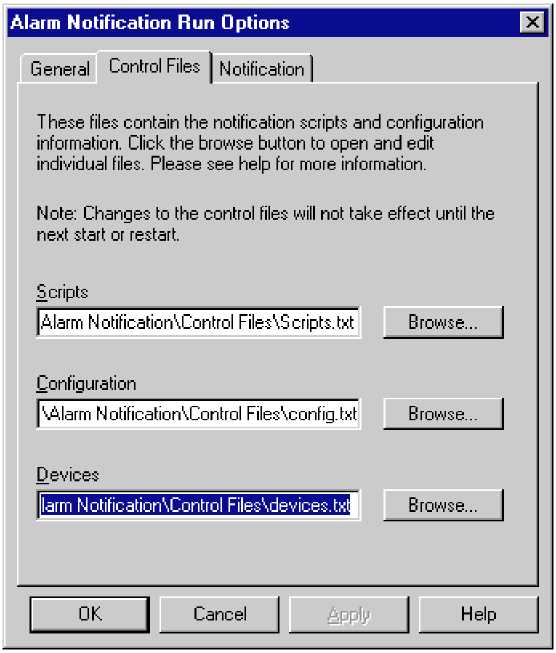
WARNING

Do not work directly in the sample Devices.txt file. This file is overwritten when MAT is reinstalled or upgraded and any changes will be lost.

2 In the Alarm Notification application, choose **Run Options** from the Configuration menu.

3 Click the **Control Files** tab. See Figure 10.

Figure 10
Alarm Notification Run Options window, Control Files tab



These files contain the notification scripts and configuration information. Click the browse button to open and edit individual files. Please see help for more information.

Note: Changes to the control files will not take effect until the next start or restart.

Scripts
Alarm Notification\Control Files\Scripts.txt

Configuration
\Alarm Notification\Control Files\config.txt

Devices
larm Notification\Control Files\devices.txt

4 Click the Browse button next to the Devices field. The Open dialog box appears. Use this dialog box to find and select the new devices file. Click **Open** to have the Notepad application open your copy of the devices file.

5 Replace the IP address following "Meridian1" with the IP address of your Meridian 1 system. You may also provide an alias.

6 For each additional Meridian 1 system or non-Meridian device to be monitored, enter a device type name, an IP address, and (optionally) an alias.

- Device type represents the type of device, for example, "XYZrouter". You use this name in the configuration file to identify SNMP traps.
- IP address or PC host file name. If the PC host file is used, the address is obtained from the PC host file.
- Aliases are alternate names you can define, which identifies each device within the Alarm Notification window.

7 Save as text and close the Notepad window.

Note: Keep a backup copy of your devices file on your local drive.

Example of device file entries listing monitored Meridian 1 systems

```
Meridian1 147.114.45.6 nmkpy716
Meridian1 147.114.45.4
Meridian1 nmkpy711 myM1
```

Configuration file

A sample configuration file is already set up for Meridian systems. Complete the following steps only if you want to monitor additional devices, such as routers or printers. Otherwise, you may skip this procedure. See "Control Files Included with Alarm Notification" on page 71.

To add configuration information:

1 In the Windows Explorer, rename a copy of the sample Config.txt file, located in the MAT directory:

```
C:\Nortel\Common Data\Alarm Notification\Control Files
```

For example, the new filename might be *my_config*.

WARNING

Do not work directly in the sample Config.txt file. This file is overwritten when MAT is reinstalled or upgraded and your changes will be lost.

2 In the Alarm Notification application, choose **Run Options** from the Configuration menu.

3 Click the **Control Files** tab. See Figure 10.

4 Click the Browse button next to the Configuration field. The Open dialog box appears. Use this dialog box to to find and select the new configuration file. Click **Open** to have the Notepad application open your copy of the configuration file

5 To add a new device, type the word *device* followed by the device name (as defined in the devices file), followed by the major and minor trap types associated with the device (refer to the device manual).

6 Below the device name, enter the following information for each event to be monitored on the device:

- Object identifier associated with the event (refer to the device manual to find this information).
- Variable type (only 'integer' and 'string' are supported).
- Variable name (you will use the variable name to refer to this event in notification scripts).
- Event name (in quotations). This name identifies the event in the Alarm Notification window.

See “Example of configuration file entries for Meridian 1 systems” on page 41.

Note: You may use the standard Meridian 1 event definitions (variable type, variable name, and event name) to define similar events for non-Meridian devices. The standard Meridian event names map the event values to corresponding fields within the Alarm Notification window and Event Properties sheet. If non-standard definitions are used, event information appears in the Additional Information field.

7 Repeat steps 5 and 6 for each non-Meridian 1 device to be monitored.

8 Save as text and close the Notepad window.

The following example shows the standard configuration file to process Meridian 1 events. Modify this file to add other systems to be managed. Users should be familiar with scripting logic and programming principles to effectively use and extend this application’s capabilities.

Example of configuration file entries for Meridian 1 systems

```
device Meridian1 6.10 {
1.3.6.1.4.1.562.3.3.7.1.0 integer $CurrentAlarmSeqNum
1.3.6.1.4.1.562.3.3.7.2.0 string $CurrentAlarmTime
1.3.6.1.4.1.562.3.3.7.3.0 integer $CurrentAlarmSeverity "Severity"
1.3.6.1.4.1.562.3.3.7.4.0 string $CurrentAlarmErrorCode "Error Code"
1.3.6.1.4.1.562.3.3.7.5.0 string $CurrentAlarmComponentId
1.3.6.1.4.1.562.3.3.7.6.0 string $CurrentAlarmComponentAddress
1.3.6.1.4.1.562.3.3.7.7.0 string $CurrentAlarmDescriptiveText "Text"
1.3.6.1.4.1.562.3.3.7.8.0 string $CurrentAlarmOperatorData "OperatorData"
1.3.6.1.4.1.562.3.3.7.9.0 string $CurrentAlarmExpertData "Expert Data"
1.3.6.1.4.1.562.3.3.7.10.0 string $CurrentAlarmCounts
}
```

Contained in the configuration file is a set of device definitions, each followed by a list of the monitored devices of that type. Each device definition begins with the word *device*, followed by the designated name, and followed by a list of traps allowed from the defined device type. For the example above, the designated name is Meridian 1, and the allowed trap values are 6 and 10.

The next lines in the configuration entries contain SNMP object identifiers, followed by a variable type, and followed by a variable name and an optional printable name in quotes. The variable name may be referenced in the scripting language and the printable name is displayed in the Network Event browser pane in the Alarm Notification window.

> ***Note:*** Only integer and string values are supported in the device definition entries.

Scripts file

The scripts file defines alarm processing and notification. A sample scripts file is provided which you can modify. See “Control Files Included with Alarm Notification” on page 71.

To set up the scripts file:

1 In the Windows Explorer, rename a copy of the sample Scripts.txt file, located in the MAT directory:

```
C:\Nortel\Common Data\Alarm Notification\Control Files
```

For example, the new filename might be *my_scripts*.

WARNING

Do not work directly in the sample Scripts.txt file. This file is overwritten when MAT is reinstalled or upgraded and any changes will be lost.

2 In the Alarm Notification application, choose **Run Options** from the Configuration menu.

3 Click the **Control Files** tab. See Figure 10.

4 Click the Browse button next to the Scripts field. The Open dialog box appears. Use this dialog box to find and select the new scripts file. Click **Open** to have the Notepad application open your copy of the scripts file.

Using the sample script as a guide, create your own notification script file. For an example of scripts files, see “Sample Alarm Notification Scripts” on page 65 and “Control Files Included with Alarm Notification” on page 71.

5 When finished, save as text and close the Notepad application.

A script includes variables, rules, notifications, functions and logical expressions that evaluate and may apply to event data. See “Scripting” on page 53. Values defined within a script are exclusive and visible only to that script. The script container is delimited by “curly” brackets **{}**.Global values can be defined that are visible to all scripts.

When the Alarm Notification application starts, each script is executed in the order defined in the script file.

Determine the MAT PC IP address

To find your PC's IP address:

1 From the Start button, select Settings - Control Panel. The Control Panel window opens.

2 Open the Network icon to display the tabbed dialog box. Click on the Configuration tab. A list of installed network components is presented.

3 Select the TCP/IP network component used by your PC. Depending on the number of installed components, you may have to scroll to see the correct component.

4 With the component selected, click on Properties. The TCP/IP tabbed window opens.

5 Click on the IP Address tab. Note the IP address shown. This is the IP address unique to this PC. You will enter this information in Overlay 117 to specify where the alarm event will be received.

6 Close all the control panel related windows and return to your desktop.

Enable alarms with Overlay 117

To enable alarms with Overlay 117:

1 Click on the System Terminal icon from the toolbar in the MAT system window. The System Terminal Selection window opens.

2 Click on the Ethernet/PPP (Overlay Passthru) radio button then click OK. The System Terminal window opens.

3 Log in with your administrator's user name and password.

Note: You must have appropriate access priviledges to use Overlay 117.

4 Load Overlay 117 by entering `ld 117` in the command line. The => prompt appears in the Command Results pane indicating the system terminal application is ready to accept your input.

5 Type `prt open_alarm` to see if other users are currently accessing the system. A list of slots currently in use is displayed. Slots are numbered from 0 through 7, for a total of eight available slots. Note the next available slot.

6 Type `set open_alarm` *`n IP_address`* where *`n`* is the next available slot number and *`IP_address`* is the IP address of your MAT PC. See “Determine the MAT PC IP address” on page 43.

Note: Assigning your IP address to a slot currently in use will disconnect that user from the system preventing them from receiving alarms information.

7 Verify the overlay has accepted your entry by typing `prt open_alarm`. The list of slots and IP addresses receiving alarms is displayed. Verify that your particular slot and IP address is included.

Note: Overlay 117 accepts abbreviations of the various commands. For example, you can type the abbreviation `prt op` instead of `prt open_alarm.`

8 Log out and close the system terminal window.

Sample Overlay 117 session

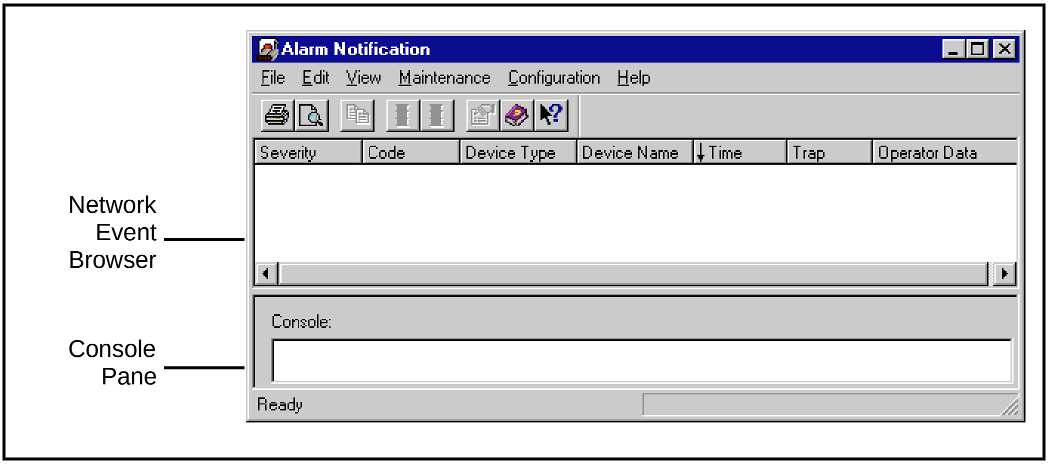
The following is a representative sample of a system terminal session using Overlay 117 to enable alarms. In this example, the MAT PC that will receive the alarms information has the IP address `47.82.40.6`. Slots 0 and 1 are already in use by other PCs. Use the next available slot 2 to enter the new MAT PC IP address. Note the => prompt used by the overlay. All IP addresses are for example purposes only. Additional information about Overlay 117 is available in the online Help

```
logi admin1
PASS?
WARNING: THE PROGRAMS AND DATA STORED ON THIS SYSTEM ARE
LICENSED TO OR ARE THE PROPERTY OF NT/BNR AND ARE LAWFULLY
AVAILABLE ONLY TO AUTHORIZED USERS FOR APPROVED PURPOSES.
UNAUTHORIZED ACCESS TO ANY PROGRAM OR DATA ON SYSTEM IS
NOT PERMITTED. THIS SYSTEM MAY BE MONITORED AT ANY TIME
FOR OPERATIONAL REASONS. THEREFORE, IF YOU ARE NOT AN
AUTHORIZED USER, DO NOT ATTEMPT TO LOGIN.
BSD000
.ld 117
OAM000
=> prt open_alarm
Open Alarm destination #0 is 47.82.40.237
Open Alarm destination #1 is 47.82.40.119
=> set open_alarm 2 47.82.40.6
=> prt op
Open Alarm destination #0 is 47.82.40.237
Open Alarm destination #1 is 47.82.40.119
Open Alarm destination #2 is 47.82.40.6
```

Using the application

Access the Alarm Notification window from the **Utilities** menu in the Navigator window. See Figure 11.

Figure 11
Alarm Notification window



The top pane of the Alarm Notification window, the Network Events browser, displays all events received since starting the application. The bottom pane, called the console, displays notifications defined in the script sent to *con*.

Consult the **Help** menu for specific descriptions of the menus, toolbar, column headings, window panes, and other functions available in the Alarm Notification window.

Use the Alarm Notification window to:

- start, stop, and restart alarm processing
- specify the control files used by the application
- view events as they are received from defined systems and devices
- view script and notification output in the console as they are received
- view a web page containing received events

Understanding events processing

Events received are displayed in the Network Events browser. As each event is received, it is placed in a queue for processing by the application. Each event is checked to see if it originates from a defined system or supported equipment. Events received from undefined systems devices are ignored and discarded from the queue.

The application executes every applicable script for each specific event type. If a rule is matched to an event type, the programmed output can either be displayed in the Alarm Notification window console pane or sent as one of the other available notification types. Output to the console occurs immediately but notifications are written to disk in an internal notification spool directory used by the application to facilitate delivery and administrative recordkeeping.

The notification process periodically polls the notification spool directory. Notifications found here are sent to the specific device or function for this notification type. Results of the notification can be displayed in the console pane. If a failure occurs due to power failure or software or hardware error, the uncompleted notifications are resumed on restart.

Viewing events properties

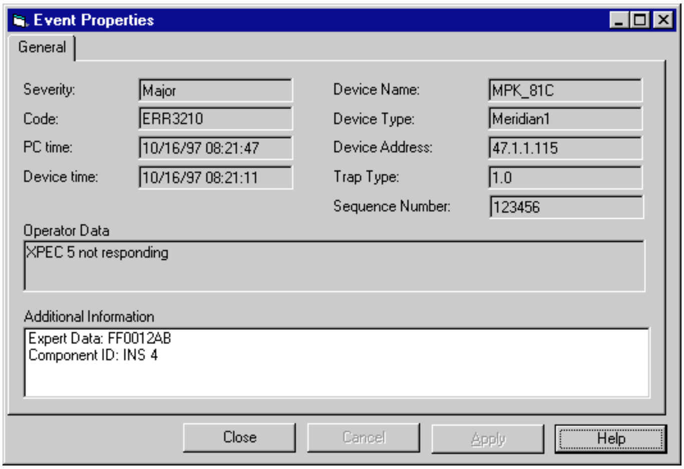
The Alarm Notification application receives events from many different systems and devices. Each event source may have different characteristics requiring the user to enter different information to fully define an event source. Use the property sheet in the Event Properties window to view an event’s values as defined in the specified configuration file.

To view the Event Properties window, select an event in the Network Browser pane and click on the **Properties** icon.

> ***Note:*** The Event Properties window is optimized for Meridian 1 events.

The Alarm Notification application processes events you have defined in the configuration file from specific systems or devices you have defined in the devices file. Events from undefined systems or devices are ignored.

Figure 12
Event Properties window



Severity: Major

Code: ERR3210

PC time: 10/16/97 08:21:47

Device time: 10/16/97 08:21:11

Device Name: MPK_81C

Device Type: Meridian1

Device Address: 47.1.1.115

Trap Type: 1.0

Sequence Number: 123456

Operator Data
XPEC 5 not responding

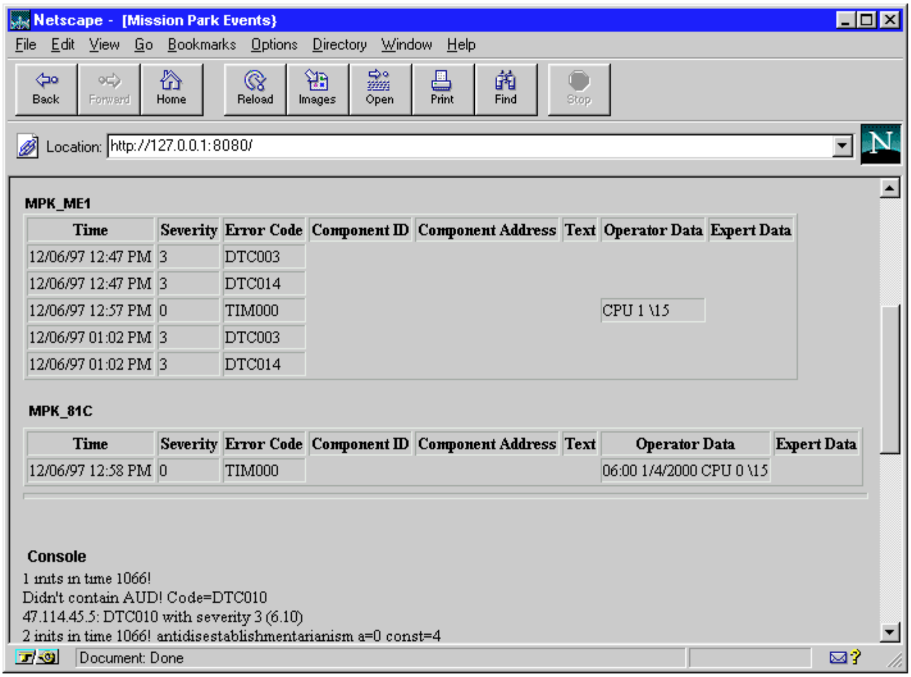
Additional Information
Expert Data: FF0012AB
Component ID: INS 4

Viewing events through a web browser

The web page display is for viewing only the list of monitored devices, events, and console output including notifications and their status. The web page does not allow scripts to be loaded, unloaded, started, or stopped. The display is configured to refresh at intervals specified in **Alarm Notification Run Options**.

Figure 13 shows an example of a web page displaying a list of events.

Figure 13
Web page view


Netscape - [Mission Park Events]
File Edit View Go Bookmarks Options Directory Window Help
Back
Forward
Home
Reload
Images
Open
Print
Find
Stop
N
Document: Done

Location: http://127.0.0.1:8080/

MPK_ME1

Time	Severity	Error Code	Component ID	Component Address	Text	Operator Data	Expert Data
12/06/97 12:47 PM	3	DTC003					
12/06/97 12:47 PM	3	DTC014					
12/06/97 12:57 PM	0	TIM000				CPU 1 \15	
12/06/97 01:02 PM	3	DTC003					
12/06/97 01:02 PM	3	DTC014					

MPK_81C

Time	Severity	Error Code	Component ID	Component Address	Text	Operator Data	Expert Data
12/06/97 12:58 PM	0	TIM000				06:00 1/4/2000 CPU 0 \15	

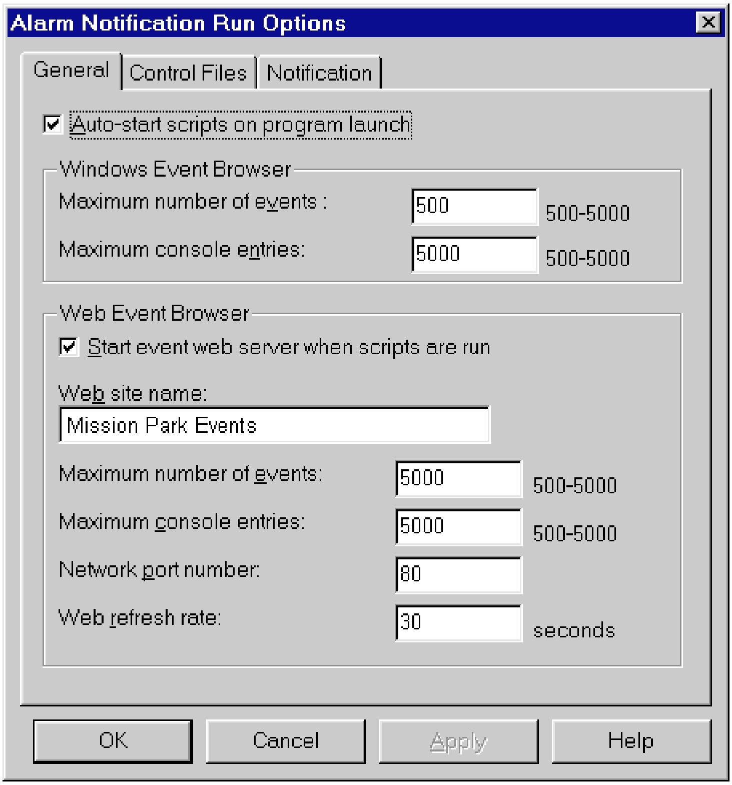
Console

```
1 inits in time 1066!
Didn't contain AUD! Code=DTC010
47.114.45.5: DTC010 with severity 3 (6.10)
2 inits in time 1066! antidisestablishmentarianism a=0 const=4
```

Setting Run Options for web browsers

Use the Run Options dialog box to define the attributes for viewing alarms information in your browser application. See Figure 14. The following describes the buttons and fields in this dialog box.

Figure 14
Setting Run Options for the web event browser



Alarm Notification Run Options

☑ Auto-start scripts on program launch

Windows Event Browser

Maximum number of events : 500 500-5000

Maximum console entries: 5000 500-5000

Web Event Browser

☑ Start event web server when scripts are run

Web site name: Mission Park Events

Maximum number of events: 5000 500-5000

Maximum console entries: 5000 500-5000

Network port number: 80

Web refresh rate: 30 seconds

- Start event web server when scripts are run

  Automatically starts the web server on this MAT PC when you Start event processing

- Web site name

Name that appears in the title bar of the Web Event server.

- Maximum number of events

  Sets the number of events stored in memory for the Web Event server. When the limit is reached, older events are replaced by new events.

- Maximum console entries

  Sets the number of entries allowed in the Console area of the Web Event server. When the limit is reached, older entries are replaced by new events.

- Network port number

  Network port assigned to the Web Event server. Web servers are usually assigned port 80 or 8080. If you already have a web server running on these ports you must assign another port to the Web Event server. Specify the new port number in the URL you enter. For example, `http://47.82.40.6:`*`n`* where *n* is a port number.

- Web refresh rate

  Rate at which the web server resends latest events to the web browser to refresh the Web Event server display.

Setting up the web event browser

To set up the web browser to display alarms information:

1 Open your web browser application.

2 Open the location of your MAT PC that is receiving alarms information. This open command can be accessed differently depending on your particular browser.

3 Enter the IP address of your MAT PC as the URL. For example, `http://47.82.40.6.` Click Open. The browser application receives the alarms information and displays it in a web page.

Note: Each URL location can be bookmarked.

Specifying which systems to view

Refer to your devices file for system names. See “Devices file” on page 36. To view a specific system in the web page:

1 Enter the IP address of your MAT PC running the Alarm Notification application as the URL, followed by a “/”, followed by the system name of the system you want to view.

For example, `http://47.82.40.6/system1.`

To view more than one system in the web page:

1 Enter the IP address of your MAT PC as the URL, followed by a “/”, followed by the system names you want to view. Separate each system name with a comma character. Do not enter spaces between system names.

For example, `http://47.82.40.6/system1,system2.`

Note: Systems are displayed in the order specified.

Scripting

Scripting involves using the syntax of the programming language in the Alarm Notification application to create text files specifying certain actions to be executed for defined events. A script includes one or more logical expressions that evaluate event data and provide notification instructions. The script file may contain many scripts. When the Alarm Notification application is started, all scripts are executed against each new event as it is received.

Scripts are executed in the order defined in the script file. To cause a script to be skipped when it is not applicable, use the *when* operator. Users should be familiar with scripting logic and programming principles to effectively use and extend this application’s capabilities.

Scripting syntax includes the following:

- Data types
- Operators
- Notifications
- Rules
- Comments
- Functions

These are described below. Consult the **Help** menu for specific examples of scripting syntax.

Data types

The scripting language supports three data types:

- counter

  Counters contain signed integer values. Counters may have values assigned to them at their time of definition. Multiple variables of the same data type may be declared in the same statement.

- timer

  Timers are counters that are automatically incremented when the time changes. Default timers increment once every minute. Specific update intervals other than the default increment may be defined.

- string

  Strings contain arbitrary alphanumeric data. A default string contains up to 80 characters. If more data is placed in a string than the string definition allows, the application truncates the entry.

Operators

Scripts usually contain a logical expression for testing event data. Logical expressions support the following operators, which may be used in any combination and with the aid of parentheses to clarify the order of operations.

The Alarm Notification application supports the following operators:

Table 2
Operators

Operator	Description
+, -, *, /	addition, subtraction, multiplication, division
<, <=	less than, less than or equal
>, >=	greater than, greater than or equal
=, !=	equal, not equal
<>	contains (look for substrings)
and, or	logical AND, logical OR
:=	assign a value to a variable. The data types must agree or a compiler error will result when the script is executed. If a value is assigned to a string value, the string must be declared large enough to contain the new value.

For an example of how some of these operators are used, see “Sample Alarm Notification Scripts” on page 65 and “Control Files Included with Alarm Notification” on page 71.

Notifications

Notifications define the message text and the means by which it is conveyed. The Alarm Notification application supports the notification types shown in Table 3:

Table 3
Notification types

Name	Definition
console	sends output to the console pane in the Alarm Notification window. This type of notification is the simplest and contains no fields.
npager, apager	sends messages to numeric (npager) or alphanumeric (apager) pagers
email	sends an electronic mail message to a remote system using Simple Mail Transfer Protocol (SMTP). For email to work correctly, an SMTP-capable host must be network accessible to the MAT PC.
modem (text)	sends message text over a modem attached to the MAT PC to a remote system, such as an alarm collection management workstation. To use this feature, a modem and a phone line must be connected, supported, and available. If the destination is busy, the Alarm Notification application will retry later or send the message to an alternate destination. Use the **Alarm Notification Run Options** dialog box to define additional actions.
snmp	sends SNMP traps to a remote system
file	saves the output to disk as a text file

Notification types

All notification types except *console* accept the fields *days* and *times* in their definitions. The *days* field may contain a quoted list of valid days (for example: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday) or a range of valid days of (for example, Sunday-Saturday). Different destinations may be specified depending on the current day, date, time, or alarm notification type.

Alarm Notification can deliver with the *send* syntax six types of notifications:

Console

Displays notification to the console panel in the Alarm Notification window. No mandatory parameters are needed.

This type of notification sends output to the Console in the Alarm Notifications window. The console notification is the simplest and contains no fields. A default console notification, *con*, is provided.

Example:

```
// (no definition necessary)
.
.
.
// send a console notification
send(con,"M1 alarm: ",
 $CurrentTrapSource," - " ,         // Name of this M1
 $CurrentAlarmErrorCode," - " ,      // M1 error code (i.e., BUG1234)
 $CurrentAlarmTime," - " ,           // Timestamp from M1
 $CurrentAlarmDescriptiveText," - ", // Text with error message
 $CurrentAlarmOperatorData);         // More text with error message
```

Note: This script displays the text “M1 alarm:” then the values for the listed traps which includes the name of the system, the error code, the time when the alarm occurred, and text associated with the error code. Note the use of *con* with the *send* syntax in the script.

Pager

Alarm Notification supports both alphanumeric and numeric pagers. You define the list of paging destinations and supported functionality for each pager. Be sure you have a modem and a valid phone line connection. Messages must match the pager type: up to 30 characters for alphanumeric pagers and only digits for numeric pagers.

Mandatory parameters are:

- `phone:=”408-555-1212”;`

  (the phone number you dial to connect to your pager service)

- `pin:=”123456”;`

  (the alpha pager PIN number, for the type *apager* only)

Example

The following example shows a script defined to call an alphanumeric pager named “my_pager” only on Mondays and Thursday through Saturday, from 9AM to 5PM. The PIN and the pager service number are specified. Note that the number needed to access an external phone line is included.

```
notification apager my_pager
{
   days:=”monday, thursday-saturday”;
   times:=”9am-5pm”;
   pin:=”123456”;
   phone:=”9,408-555-1212”;
}
```

> ***Note:*** The indentations facilitate reading the script and do not affect how the script is interpreted by the application.

E-mail

Write a message that the application sends to a specified list of recipients. The application uses Simple Mail Transfer Protocol (SMTP) to send the message. Be sure you have an SMTP-capable host connected and accessible to the MAT PC for this notification to work correctly. This host must be configured in MAT before this feature is activated. Each *send* statement is treated as a separate e-mail message.

Email messages coming from Alarm Notification are sent with “MAT” as the sender. The receiving email program may try to match MAT with a corresponding local user account and displays the closest match in the email’s header. Although the header information may not be accurate due to mismatches between the term “MAT” and the local user account information, the email message is still displayed as defined by the notification.

Mandatory parameters are:

- `address:=”support@tech.com”;`

  (the recipient’s valid e-mail address)

- `server:=”192.9.200.1”;`

  (IP address or hostname of SMTP mail server)

Example

```
// define an email notification
   notification email my_email {
      address:="joe@acme.com";
      server:="192.9.200.1";
  }
.
.
.
// send an email message
send (my_email,$CurrentTrapSource,"-MPK alarm:",
$CurrentAlarmErrorCode);
```

Note: This script defines an e-mail notification named *my_email* which has an associated recipient address and the mail server IP address. This script sends e-mail to the address stating the source of the alarm and the alarm error code. Note the use of the named email notification *my_email* with the *send* syntax in the script.

Text (over modem)

Write a message that the application sends to a remote modem. The application connects to the remote modem (usually a remote support site) defined by the user and transmits the message. Be sure you have a modem and a valid phone line connection.

If the remote modem is busy, the application stores and notes the message for a later delivery or sends it to another specified alternate destination.

Mandatory parameters are:

- `phone:=”408-555-4321”;`

  (the phone number you dial to connect to the remote modem)

Example

```
//define text over modem notification
   notification modem tech_center   {
   phone:="9,555-4321";
  }
.
.
.
//send text over modem notification
send (tech_center,$CurrentTrapSource,"-MPK alarm:",
$CurrentAlarmErrorCode);
```

Note: This script defines a modem notification named *tech_center* which has an associated number to dial to connect to the remote modem. This script sends text to the dialed remote modem stating the source of the alarm and the alarm error code. Note the use of the named modem notification *tech_center* with the *send* syntax in the script

SNMP

The MAT PC can generate Simple Network Management Protocol (SNMP) traps as defined by the user. Define SNMP Object Identifiers (OIDs) as parameters in the *send* syntax. Specify a list of TCP/IP addresses or registered hostnames on the Ethernet network to receive the trap. However, receipt is not guaranteed once messages are transmitted. You should be familiar and knowledgeable with SNMP to fully use this notification type.

Mandatory parameters are:

- `address:=”192.9.200.1”;`

  (IP address or hostname of the destination for the trap)

- `trap:=”6.10”;`

  (trap type in Major type.Minor type format)

Example

Below is an example, where *control_center* is already defined as an SNMP notification:

```
// define an SNMP notification
   notification snmp control_center {
      address:="192.9.200.1";
      trap:="6.10";
   }
.
.
.
// send an SNMP notification to the control center with the same trap
//format as the M1

   send(control_center,
"1.3.6.1.4.1.562.3.3.7.1.0","Integer",$CurrentAlarmSeqNum,
"1.3.6.1.4.1.562.3.3.7.2.0","OctetString",$CurrentAlarmTime,
"1.3.6.1.4.1.562.3.3.7.3.0","Integer",$CurrentAlarmSeverity,
"1.3.6.1.4.1.562.3.3.7.4.0","OctetString",$CurrentAlarmErrorCode,
"1.3.6.1.4.1.562.3.3.7.5.0","OctetString",$CurrentAlarmComponentId,
"1.3.6.1.4.1.562.3.3.7.6.0","OctetString",$CurrentAlarmComponentAddress,
"1.3.6.1.4.1.562.3.3.7.7.0","OctetString",$CurrentAlarmDescriptiveText,
"1.3.6.1.4.1.562.3.3.7.8.0","OctetString",$CurrentAlarmOperatorData,
"1.3.6.1.4.1.562.3.3.7.9.0","OctetString",$CurrentAlarmExpertData);
```

Note: The script identifies the value type of the trap generated, whether an integer or an octet string.

Log file

Write a message that the application saves to a text file. Specify the storage location of this text file in the script. If no pathname is specified, the file is saved in the same directory as the Alarm Notification application.

Mandatory parameters are:

- `filename:=”c:\Normat\sample_log”;`

  (full pathname and filename)

Example

```
// define a file notification
notification file sample_file {
filename  :=  "c:\eventlog.txt";
}
.
.
.
// send a file notification
send(sample_file, “M1 alarm: “,
$CurrentTrapSource," - " ,         // Name of this M1
$CurrentAlarmErrorCode," - " ,      // M1 error code (i.e., BUG1234)
$CurrentAlarmTime," - " ,           // Timestamp from M1
$CurrentAlarmDescriptiveText," - ", // Text with error message
$CurrentAlarmOperatorData);         // More text with error message
```

Note: This script defines a log file notification named *sample_file* having an associated pathname defining a storage location on the c: drive. The store file is named *eventlog.txt*. Note the use of the named file notification *sample_file* with the *send* syntax in the script

Rules

Rules allow users to define actions that may be applied to a given event. Rules may only be defined in scripts. By default, rules are examined in a top-down manner. An *infer* statement may be used to explicitly specify the order in which rules are examined. A rule consists of an *if* statement containing a logical expression, followed by an instruction.

A rule may also include an *else* statement, which is only executed if the logical expression in the *if* statement is false. Within a rule, a *send* statement or a function may be called. New variables may not be defined within the scope of a rule.

Comments

Comments provide a convenient way of adding informational notes within a script. To include comments within a script use the C conventions (begin with /* and end with */) or C++ conventions (begin the comment with //).

For example:

```
/* This is a comment. */
// This is another comment.
```

Note: Many of the scripts presented in this user guide have portions noted as comments. Remove the comment tags for the application to interpret these as actual scripting code.

Functions

Functions contain a combination of logical expressions and optional variable declarations. They may accept parameters and return a single result. Functions may be invoked within logical expressions, or rules, or invoked within themselves. Within a function, assignments may occur along with *if* and *loop* statements.

Sample Alarm Notification Scripts

This section shows portions of a sample script to illustrate scripting syntax that perform common alarm notification tasks.

Log file

This script uses the *file* notification. This script saves all events to the filename “sample_log.txt” in the defined location.

> ***Note:*** Windows “long” file names are allowed in the scripts but are truncated when the system saves the file. Keep your file names to the standard eight character length.

```
notification file sample_file {
   filename:="c:\sample_log.txt";
}
```

Numeric pager

This script uses the *npager* notification. Customize this script by typing in your numeric pager number

```
notification npager sample_numeric_pager {
   phone:="9,555-1212";
}
```

Alphanumeric pager

This script uses the *apager* notification. Customize this script by typing in your alphanumeric pager number and Personal Identification Number (PIN).

```
notification apager sample_alpha_pager {
   phone:="9,555-1212";
   pin:="101565";
}
```

Severity code descriptions

This script examines error codes generated by the Meridian 1 and assigns descriptive text to them. This portion of the script uses the *counter* data type and the *send* alarm notification to send alarm notifications to a file and a pager. Note the use of *rule* and *send* syntax. Note the *$Current...* references to configuration file entries. See “Example of configuration file entries for Meridian 1 systems” on page 41.

```
/* Provide variables to map M1 severity values into words */
counter info:=0,minor:=1,major:=2,critical:=3,debug:=4;
script SampleScript {
   counter bug3456:=0;

   /* This rule looks for critical M1 events */
   rule check_critical {
      if ($CurrentTrapDevice="Meridian1" and
$CurrentAlarmSeverity=critical)
```

Note: The *if* statement ensures that only critical alarms from the Meridian 1 are processed.

```
         {
         /* print event to console */
         send(con,"M1 alarm: ",
    $CurrentTrapSource," - " ,         // Name of this M1
   $CurrentAlarmErrorCode," - " ,      // M1 error code (i.e., BUG1234)
   $CurrentAlarmTime," - " ,           // Timestamp from M1
   $CurrentAlarmDescriptiveText," - ", // Text with error message
   $CurrentAlarmOperatorData);         // More text with error message
```

Note: The script displays on the console pane information about the M1 alarm, including the system where the alarm originated, the error code of the alarm, the time of the alarm, any descriptive text associated with the alarm, and other text associated with the error alarm.

```
         /* append event to log file */
         send(sample_file,"M1 alarm: ",
    $CurrentTrapSource," - " ,         // Name of this M1
   $CurrentAlarmErrorCode," - " ,      // M1 error code (i.e., BUG1234)
   $CurrentAlarmTime," - " ,           // Timestamp from M1
   $CurrentAlarmDescriptiveText," - ", // Text with error message
   $CurrentAlarmOperatorData);         // More text with error message
```

Note: The script saves to a named file on disk the same information about the M1 alarm that was displayed on the console.

```
         /* optionally send message to alpha pager */
send(sample_alpha_pager,$CurrentTrapSource,":",$CurrentAlarmCode,"!");
*/
         }
   }
```

Note: The send command contacts the pager named as *sample_alpha_pager* with the error information “M1 : BUG1234” where M1 is the name of the system that has the error and BUG1234 is the error type.

Specific system events

This script examines error codes generated by the Meridian 1 for a specific event code and counts the number of occurrences for this event. For this example, BUG3456 is the specific event code. This portion of the script displays to the console the time when the error occurred. Customize this script by typing in your error code. This script may be of use if you are trying to troubleshoot the system for a specific problem.

This rule is named *check_specific_event* and examines events from the device named Meridian1 for error code BUG3456. If this event is detected, the console displays “Found BUG3456 at <*alarm time*>” where <*alarm time*> is the timestamp provided by the system.

This script uses the *rule* syntax.

```
rule check_specific_event {
      if ($CurrentTrapDevice="Meridian1" and
$CurrentAlarmErrorCode="BUG3456")
         {
         send(con,bug3456,")
Found ",$CurrentAlarmErrorCode," at ",$CurrentAlarmTime);
         bug3456:=bug3456+1;
         }
   }
```

Combining scripts

Several scripts are often found in a single script file. The sample scripts in this section are combined into a single text file named *Script.txt* included with the application. See “The scripts file” on page 72. Note the use of comments to document the various portions of the script.

Scripting notes

The scripting language available with Alarm Notification allows tremendous flexibility and functionality in defining how the application processes events from connected systems. You can use any text editor such as Notepad to write your script. Use the Control Files tab in **Alarm Notification Run Options** to specify the script and other control files you will use.

Customized scripts are interpreted by the Alarm Notification application. Errors in the script are noted and related error messages are displayed in the console panel in the Alarm Notification window. Scripting error messages include the line number where the error occured, as counted from the top of the text file, as well as a short error description.

Appendix A: Control Files Included with Alarm Notification

This section displays the control files that are loaded into the MAT PC during when the Alarm Notification application is installed. These files are loaded into a default location `C:\Nortel\Common Data\Alarm Notification\Control Files` unless otherwise specified.

WARNING

Always use a copy of these files when customizing them for your specific environment. Do not work directly in these files. These files are overwritten when MAT is reinstalled or upgraded and any changes will be lost.

The devices file

The following is a printout of the *Devices.txt* file included with the application.

```
# This file contains a list of specific devices to be monitored by
# Alarm Notification.  As this file may be replaced during a software
upgrade,
# it is suggested that any changes be made in a copy and the copy used.
# The following are example definitions:
#
#Meridian1 192.9.200.1 my_m1
#Meridian1 192.9.200.2
#Meridian1 sample_m1
#
# User provided devices should be added below this line.
```

The configuration file

The following is a printout of the *Config.txt* file included with the application.

```
# The following definitions contain device definitions for Nortel
# supported devices. As this file may be replaced during a software
upgrade,
it is suggested that any changes be made in a copy and the copy used.

device Meridian1 6.10 {
1.3.6.1.4.1.562.3.3.7.1.0 integer $CurrentAlarmSeqNum
1.3.6.1.4.1.562.3.3.7.2.0 string $CurrentAlarmTime
1.3.6.1.4.1.562.3.3.7.3.0 integer $CurrentAlarmSeverity "Severity"
1.3.6.1.4.1.562.3.3.7.4.0 string $CurrentAlarmErrorCode "Error Code"
1.3.6.1.4.1.562.3.3.7.5.0 string $CurrentAlarmComponentId
1.3.6.1.4.1.562.3.3.7.6.0 string $CurrentAlarmComponentAddress
1.3.6.1.4.1.562.3.3.7.7.0 string $CurrentAlarmDescriptiveText "Text"
1.3.6.1.4.1.562.3.3.7.8.0 string $CurrentAlarmOperatorData "Operator
Data"
1.3.6.1.4.1.562.3.3.7.9.0 string $CurrentAlarmExpertData "Expert Data"
1.3.6.1.4.1.562.3.3.7.10.0 string $CurrentAlarmCounts
}

# Add user supplied device definitions below this comment line.
```

The scripts file

The following is a printout of the *Scripts.txt* file included with the application.

```
/* This file contains a simple example of script file usage. */

/* This is a sample definition for using a log file.  All events sent
   to this notification will be appended to the filename defined below.
   Please note that Windows "long" file names are not supported. */

notification file sample_file {
   filename:="c:\sample_log.txt";
}
```

```
/* This is a sample definition for using a numeric pager
   To use, the phone number should be changed to your pager number
   and the notification (as well as the references to it) should be
   uncommented. */
/*
notification npager sample_numeric_pager {
   phone:="9,555-1212";
}
*/

/* This is a sample definition for using a numeric pager
   To use, the phone number should be changed to your pager number,
   your PIN number should be added, and the notification (as well
   as the references to it) should be uncommented. */
/*
notification apager sample_alpha_pager {
   phone:="9,555-1212";
   pin:="101565";
}
*/

/* Provide variables to map M1 severity values into words */
counter info:=0,minor:=1,major:=2,critical:=3,debug:=4;

script SampleScript {
   counter bug3456:=0;

   /* This rule looks for critical M1 events */
   rule check_critical {
      if ($CurrentTrapDevice="Meridian1" and
$CurrentAlarmSeverity=critical)
         {
         /* print event to console */
         send(con,"M1 alarm: ",
    $CurrentTrapSource," - " ,         // Name of this M1
   $CurrentAlarmErrorCode," - " ,      // M1 error code (i.e., BUG1234)
   $CurrentAlarmTime," - " ,           // Timestamp from M1
   $CurrentAlarmDescriptiveText," - ", // Text with error message
   $CurrentAlarmOperatorData);         // More text with error message

         /* append event to log file */
         send(sample_file,"M1 alarm: ",
    $CurrentTrapSource," - " ,         // Name of this M1
   $CurrentAlarmErrorCode," - " ,      // M1 error code (i.e., BUG1234)
   $CurrentAlarmTime," - " ,           // Timestamp from M1
   $CurrentAlarmDescriptiveText," - ", // Text with error message
   $CurrentAlarmOperatorData);         // More text with error message

         /* optionally send message to alpha pager */
/*

send(sample_alpha_pager,$CurrentTrapSource,":",$CurrentAlarmCode,"!");
*/
         }
   }

   /* This rule looks for and counts a specific M1 event type */
   rule check_specific_event {
      if ($CurrentTrapDevice="Meridian1" and
$CurrentAlarmErrorCode="BUG3456")
         {
         send(con,bug3456,") Found ",$CurrentAlarmErrorCode," at
",$CurrentAlarmTime);
         bug3456:=bug3456+1;
         }
   }
}
```

Meridian Administration Tools

Alarm Management

User Guide

Information is subject to change without notice. Nortel Networks reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant. This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC rules, and the radio interference regulations of Industry Canada. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at their own expense.

Publication number: P0906946
Document release: Standard
Date: April 2000
Printed in the United States of America